e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

Development of Website-Based Outsourcing Services Negotiation System Using Extreme Programming Method at PT. XYZ

Dinar Rahayu[1)], Encep Riswan Suherlan[2)]
[1)] Sistem Informasi, Ilmu Terapan dan Sains, Institut Pendidikan Indonesia Garut
[2)] Sistem Informasi, Manajemen Informatika dan Komputer, STMIK Bandung
Email: dinarrah2405@gmail.com; encepriswan@gmail.com.

Accepted:	Accepted After Revision:	Published:
11 July 2025	19 August 2025	27 August 2025

Abstract

This study develops a Website-Based Outsourcing Services Negotiation System for PT XYZ, addressing the company's reliance on manual communication and document exchange, which previously caused delays and data inconsistencies. The system streamlines client registration, service submission, negotiation, and invoice generation. Extreme Programming (XP) was selected for its iterative adaptability to evolving user requirements, focusing on rapid prototyping, continuous feedback, and incremental improvement. An object-oriented design approach ensures modularity and maintainability, while black box testing validates functional accuracy. After deployment, negotiation process time was reduced by 35%, communication response time by 42%, and documentation completeness reached 100%, significantly enhancing operational efficiency, transparency, and record reliability. These measurable improvements demonstrate the system's role in accelerating PT XYZ's digital transformation and establishing a more structured, evidence-driven outsourcing negotiation process.

Keywords: Outsourcing Service, Website-Based System, Extreme Programming, Object Oriented Programming, Black Box Testing

1 INTRODUCTION

In the era of digital transformation, the adoption of digitalization in business processes has become an inevitable need for companies looking to improve operational efficiency, information transparency, and service acceleration. One sector that has been significantly impacted by this development is the outsourcing service industry. Although the problem statement outlines the general need for a digital outsourcing negotiation system, the specific context at PT XYZ reveals more pressing operational challenges. The company currently relies on manual communication channels such as email, phone calls, and in person meetings for negotiation processes, resulting in prolonged approval times, inconsistent documentation, and difficulties in tracking negotiation history.

Additionally, the absence of a centralized digital platform leads to frequent miscommunication between departments, reduced transparency in pricing discussions, and increased administrative workload. These inefficiencies hinder the company's responsiveness to client requests and limit its ability to compete with industry peers that have already adopted integrated digital solutions (Huang et al., 2021). Addressing these business and operational constraints forms a clear research gap, underscoring the practical relevance of developing a website based outsourcing services negotiation system tailored to PT XYZ's needs. One effective solution to overcome this problem is the development of a web-based negotiation information system. Such a system can simplify the entire process, from client registration and negotiation submission to service data processing and invoicing.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

A web-based platform offers accessibility, time efficiency, and can be developed with an interface customized to the user's needs (Setyawati, 2021).

Adopting an appropriate system development methodology is essential to ensure the resulting software effectively meets user needs. Extreme Programming (XP), one of the Agile methods, offers an iterative and flexible approach that is highly suitable for systems requiring close user collaboration and frequent adjustments (Fatoni & Irawan, 2019). In the context of a web-based outsourcing negotiation system, XP's short iterations, continuous testing, and rapid adaptability are advantageous in addressing dynamic requirements. Practices such as daily stand ups and sprint reviews help minimize communication gaps in dispersed projects, while user story mapping and backlog grooming can be automated to manage priorities effectively. Furthermore, continuous integration ensures that code development remains aligned with negotiated agreements (Lamada et al., 2023).

By integrating the adaptive strengths of Extreme Programming (XP) with a web-based platform, this study introduces a negotiation system tailored to real-time outsourcing collaboration. Unlike conventional approaches, it directly addresses dynamic communication needs, enhancing operational efficiency while contributing a practical model for continuous digital transformation in service-based processes.

2 LITERATURE REVIEW

The literature review in this study serves as a theoretical and conceptual basis to support the development of a website-based outsourcing service negotiation system using the Extreme Programming (XP) method. It critically examines prior works to identify relevant findings, limitations, and opportunities that this research aims to address.

2.1 Digital Transformation

Digital transformation has become a key driver of innovation in information systems, where technology is used to create new value and improve operational efficiency. According to (Verhoef et al., 2021), emphasize that transformation is not only about digitizing processes, but also about building integrated and adaptive digital systems. In the context of outsourcing services, digitization enables faster communication and higher transparency between clients and providers. Prior works, such as (Setyawati, 2021), have demonstrated the advantages of web-based systems in providing real-time access and systematic documentation. However, most of these studies focus on general service management without detailing negotiation-specific workflows or the integration of agile methods. This study addresses that gap by combining web-based platforms with an agile methodology tailored to outsourcing negotiations.

2.2 Negotiation in an Outsourcing System

Negotiation is central to establishing agreements in outsourcing, covering service scope, pricing, performance standards, and contract terms (Hou et al., 2021). Previous studies highlight the importance of strategic preparation and transparent communication in achieving a win–win outcome. While these works describe the theoretical and procedural aspects of negotiation, they often lack concrete technological implementations that support real-time, traceable negotiations. This research advances the field by embedding these negotiation principles into a digital platform, enabling structured interactions and automatic documentation.

2.3 Extreme Programming (XP)

Extreme Programming (XP) is known for short iterations, close user collaboration, and rapid adaptability (Shrivastava et al., 2021). This method has proven to be effective in adaptive and efficient software development, especially on projects that require rapid response to changing user

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

needs (Widhiastuti et al., 2023). Prior implementations, such as (Lamada et al., 2023), have successfully applied XP in web based project management, improving responsiveness to changing requirements. However, these applications generally target broader project management contexts and do not address the specific needs of outsourcing negotiations. This study builds on these strengths by adapting XP practices daily stand ups, sprint reviews, and continuous integration to reduce communication gaps and ensure alignment between negotiated agreements and implemented system features.

2.4 Object-Oriented Programming

Object-Oriented Programming (OOP) promotes modularity, reusability, and maintainability (Issyatirrahim et al., 2024). Previous research demonstrates that OOP facilitates scalable designs and easier error detection. In the context of negotiation systems, however, OOP's potential for modular workflow handling has rarely been explored. This research leverages OOP to design reusable modules for user management, negotiation handling, and documentation, enabling a flexible and secure role-based access structure.

2.5 Unified Modeling Language (UML)

UML diagrams provide a standardized way to visualize system requirements and design (Siska Narulita et al., 2024). While prior works have demonstrated UML's value in system design, few have shown its integration with agile methods for iterative refinement. This study applies UML not only for initial modeling but also as a living document updated through XP iterations, ensuring that system design evolves alongside stakeholder feedback.

2.6 Black Box Testing

Black Box Testing evaluates system functionality from the user's perspective without inspecting the source code (Putri et al., 2024). To ensure reliability, testing was performed on all core modules with a defined number of test cases, repeated at least three times under varying input conditions, and considered successful if 100% of critical cases passed without errors (Sitio et al., 2023). This approach was applied in each XP iteration to detect and resolve discrepancies early.

2.7 Synthesis and Research Gap

From the reviewed literature, several patterns emerge:

1. Digital transformation and web-based systems improve efficiency and transparency, but prior studies seldom focus on outsourcing negotiations.
2. Negotiation theory is well-documented, yet its integration into adaptive, real-time digital tools remains limited.
3. XP and OOP have been applied in various contexts, but their combined use for negotiation-specific workflows is rare.
4. UML and Black Box Testing are established tools, but their iterative application within negotiation systems is underexplored.

By addressing these gaps, this study contributes both theoretically and practically: it offers a negotiation system specifically tailored to outsourcing services, integrates agile and modular development principles, and provides an iterative, test-driven framework that ensures the system evolves in step with business needs.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

3 RESEARCH METHODS

This research adopts the Extreme Programming (XP) approach as the primary methodology to develop a web based outsourcing service negotiation system at PT XYZ. XP was selected over other Agile methods, such as Scrum and Kanban, due to its strong emphasis on short development cycles, continuous feedback, and direct collaboration with end-users. These characteristics align with PT XYZ's operational challenges, which include frequent changes in negotiation terms, the need for rapid iteration to accommodate client feedback, and the importance of maintaining traceable documentation in a dynamic outsourcing environment.

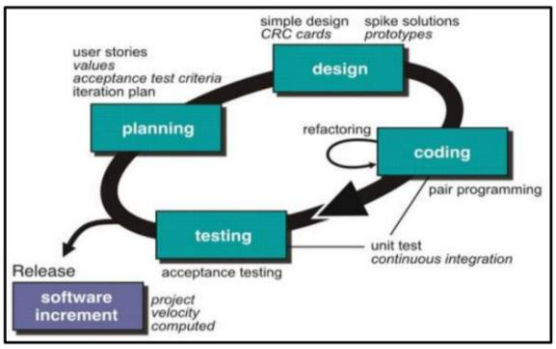

Figure 1. SDLC Extreme Programming

What follows is an explanation of each stage of extreme programming:

1. Planning
   In the planning game, developers and stakeholders collaborated to identify and prioritize system requirements in the form of user stories. For the negotiation system, high-priority stories included client registration, service request submission, negotiation revision handling, and invoice generation. This stage ensured that the system scope directly reflected the critical functions required to address PT XYZ's negotiation workflow inefficiencies.
2. Design
   System design translated user stories into UML diagrams, including Use Case, Sequence, and Class Diagrams, to represent workflows and system structure. The negotiation module, for example, was designed with flexible iteration loops to support back-and-forth revisions between clients and operations managers, a feature identified as essential in stakeholder interviews. Simple design principles were applied to maintain clarity and adaptability.
3. Coding
   The coding phase implemented features according to the agreed design, using PHP for rapid, maintainable development. Pair programming was applied to the negotiation and invoice modules, as these were mission-critical features that required both accuracy and reliability. This approach improved code quality and allowed for early detection of logic discrepancies, especially in price calculation and approval workflows.
4. Testing
   Testing was conducted using the Black Box Testing method, focusing on verifying functional behavior against defined acceptance criteria. For the negotiation system, test cases included verifying price revision handling, invoice generation accuracy, and role-based access control. Each test case was executed at least three times under different input conditions, with a success threshold of 100% for all critical cases. Any failures triggered immediate corrective actions within the same XP iteration.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

4 RESULTS AND DISCUSSION

The development of the Website-Based Outsourcing Service Negotiation System at PT XYZ was conducted in accordance with the principles of Extreme Programming (XP) methodology. This section not only reports the procedural steps taken but also provides a comparative analysis of the system's capabilities against prior studies, highlighting its novelty and contributions to both practice and literature.

4.1 User Stories and Functional Requirements

These user stories were prioritized to focus on modules critical to PT XYZ's operations, such as multi-iteration negotiation handling, automated documentation, and integrated invoicing. Unlike (Setyawati, 2021), whose work addressed general service management, this study's user stories specifically addressed real-time negotiation loops and traceability, filling a gap in the literature on outsourcing-specific systems. The main actors identified in this system are:

1. Client (who submits outsourcing service requests),
2. Administrator (who manages services, responses, and invoices),
3. Operations Manager (who oversees negotiation and approval flow).

Table 1. List of user stories

No	User Stories
1	As a client, I want to register and log in so that I can access the negotiation system.
2	As a client, I want to submit a service request so that I can initiate outsourcing negotiations
3	As a client, I want to view and edit my request before finalizing it.
4	As an admin, I want to receive service submissions and respond with an initial offer.
5	As an operations manager, I want to approve or decline negotiation submitted.
6	As a client, I want to receive notifications and reply to offers so that the negotiation process is interactive.
7	As a client, admin, and operations manager. I want to see the negotiation history for transparency.
8	As an admin, I want to generate and send invoices once the negotiation is completed.
9	As a client, I want to download the invoice as proof of agreement.

After making a list of user stories, then determine the functional requirements of this system. The functional requirements determined are as follows:

Table 2. Functional Requirements Table

No	Functional Requirement	Description
FR1	User Authentication	The system must support client, admin and operatons manager login with role-based access.
FR2	Service Request Form	Clients must be able to fill and submit a service negotiation form (with date, type of service, description).
FR3	Response Module	Operations manager must be able to view requests and reply with service terms and prices.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

No	Functional Requirement	Description
FR4	Negotiation Threading	Both parties can send and receive messages/offers in a structured conversation thread.
FR5	Notification System	The system should send email or in-app alerts for status changes.
FR6	Invoice Generation	Admin can issue invoices after both parties agree.
FR7	Document Management	Clients can download or archive submitted requests and invoices.
FR8	Monitoring	Admin can view a summary of ongoing negotiations and statuses.

This structured set of user stories and functional requirements ensured that the system addresses real business processes and supports interactive and traceable negotiation flows, rather than just static submissions.

4.2 System Development Process

At this initial stage, system requirements were collected in the form of user stories through interviews and discussions with key stakeholders, including the outsourcing service manager and administrative staff.

1. Planning
   These user stories were compiled into a product backlog and prioritized according to business importance and technical feasibility. This stage aimed to define clear functional expectations, such as: Client registration and login, Service submission, Service period selection, Cost estimation (RAB), Negotiation, Invoice generation and notification. This approach mirrors Agile best practices in (Lamada et al., 2023) but differs in its targeted prioritization of negotiation workflows essential in addressing PT XYZ's documented issues of prolonged approval times and miscommunication.
2. Design Stage
   The negotiation module was explicitly designed with back and forth revision loops to accommodate multiple agreement cycles, an aspect absent in prior general project management systems.

a. Use case diagram
   The Use Case Diagram below illustrates the interaction between three main external actors and the Outsourcing Services Negotiation System, namely the Client, Operations Manager, and Administrator. This diagram is organized to represent the main business processes that occur in digitally applying for and negotiating outsourcing services.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

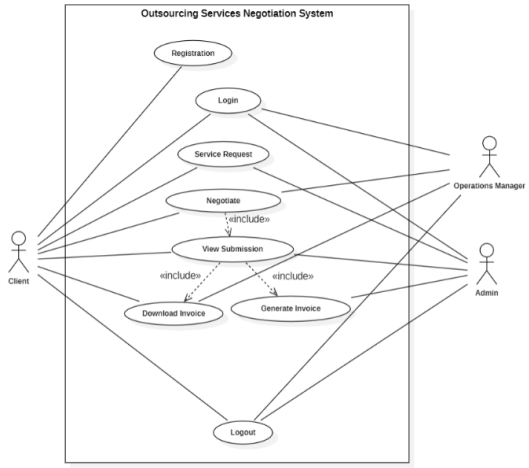

Figure 2. Use case diagram

Overall, this diagram shows how the developed system is able to facilitate the process of applying, negotiating, and administering outsourcing services in an integrated manner, while supporting the flexibility of user roles in a safe and efficient digital service ecosystem.

To understand the functional requirements and interactions between users and the system, a use case model was developed to describe how each actor communicates with the system components. The system is designed to facilitate end-to-end outsourcing service negotiations in a structured and traceable manner. Table III outlines the primary use cases identified in the system, along with the respective actors involved and a brief description of each function. This modeling approach helps define the scope of the system and ensures that all business processes are covered effectively.

The use case descriptions presented in the table illustrate the core activities performed by each actor, including clients, administrators, and operations managers. These activities range from user authentication to service submission, negotiation, and document handling. The inclusion of shared components such as the View Submission use case within multiple workflows highlights the modularity and reuse of system functions. This comprehensive model serves as the foundation for further design phases, including interface prototyping and system implementation.

Table 3. Use case description

No	Use Case	Actor	Description
1	Registration	Client	Allows new clients to create an account and gain access to the system for submitting outsourcing service requests
2	Login	Client, Administrator, Operations Manager	Authenticates users before they can access system features. Each actor has role-based access rights.
3	Service Request	Client, Administrator	The client submits a service request by filling out a form with service details. The administrator receives and verifies the request as part of the negotiation process.
4	Negotiate	Client, Administrator, Operations Manager	A structured negotiation process involving price, service scope, and delivery time, in which all actors participate to reach an agreement.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

No	Use Case	Actor	Description
5	View Submission	Client, Administrator, Operations Manager	Displays submitted data, status updates, and negotiation history. This use case is included in multiple other use cases via <<include>> relationships.
6	Generate Invoice	Administrator	The administrator creates and uploads an invoice based on the agreement reached in the negotiation. This use case includes View Submission.
7	Download Invoice	Client, Operations Manager	Allows users to access and download the final invoice as an official document for payment or archiving. This use case includes View Submission.
8	Logout	Client, Administrator, Operations Manager	Ends the current user session and logs the user out securely from the system.

b. Activity diagram

To further illustrate the system's workflow, an activity diagram was developed to represent the sequence of actions and decision points involved in processing an outsourcing service request. This diagram visualizes the interactions among the system components and actors starting from client registration through negotiation and ending with invoice generation. By mapping out the flow of activities, the diagram provides insight into the dynamic behavior of the system and supports the identification of potential bottlenecks or redundancies.

This diagram illustrates the sequential flow of user registration and login activities. The process begins with the client opening the respective page and inputting data. The system then performs validation, determines whether the input is valid or invalid, and provides appropriate feedback. If valid, the user either completes the registration or is redirected to the dashboard upon successful login.

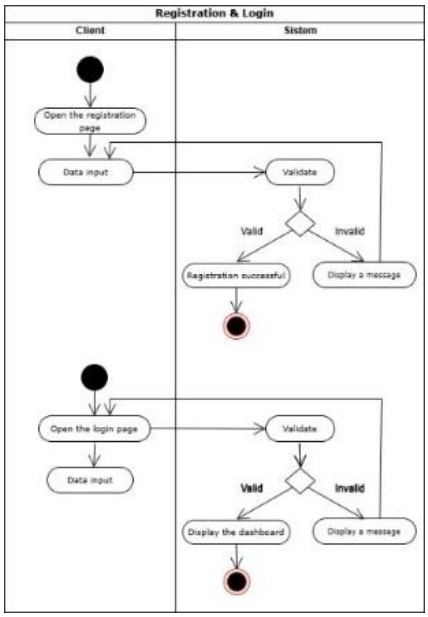

Figure 3. Activity diagram resgistration & login

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

After the authentication process is completed through login or registration, the next primary interaction performed by the client is submitting a service request. This function serves as the entry point for initiating the outsourcing service process. To represent this interaction, an activity diagram for the Service Request process is presented, outlining the sequence of activities carried out by the client, system, and administrator. The diagram illustrates how the system receives input, validates request data, and manages approval or revision feedback.

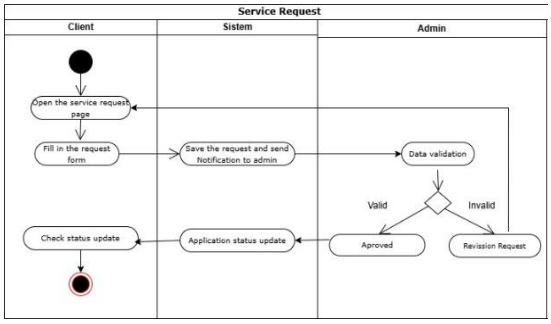

Figure 4. Activity diagram service request

Once a service request has been submitted and approved by the administrator, the process moves into the negotiation phase. This phase is essential for aligning expectations between the client and the operations manager regarding the scope, duration, and pricing of the requested service. It begins with the Operations Manager opening the negotiation page and filling out the RAB form. The system sends the service details and RAB to the client, who then reviews and either processes the terms or submits a revision. If a revision is received, the manager checks it and decides whether to approve or decline. The loop continues until a final agreement is reached.

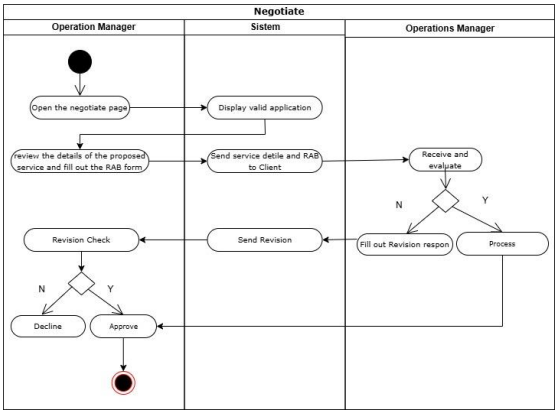

Figure 5. Activity diagram negotiate

After the negotiation process has reached a mutual agreement, the system proceeds to the final administrative stages, including viewing the finalized submission, generating the invoice, and allowing users to download the invoice as official documentation. The following activity diagram illustrates this post-negotiation workflow, showing how the client, administrator, and operations manager interact with the system to complete the service transaction.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

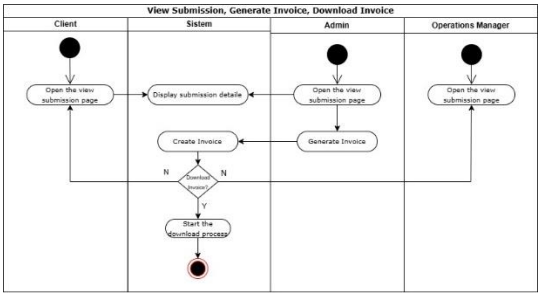

Figure 6. Activity diagram view submission, generate voice, and download invoice

After completing all primary interactions in the system including submission review, invoice generation, and document download the final step for each actor is to securely end their session. This is achieved through the logout process, which ensures data confidentiality, prevents unauthorized access, and marks the end of system usage. The following activity diagram illustrates the logout process performed by all users upon completing their tasks.

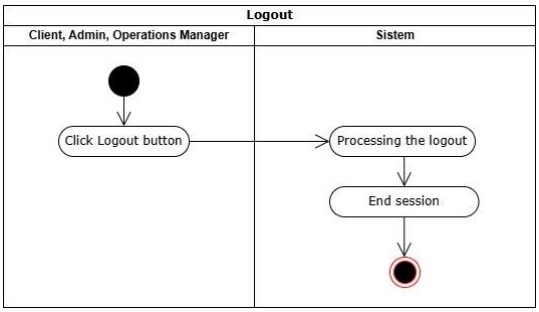

Figure 7. Activity diagram logout

The series of activity diagrams presented in this section illustrates the end-to-end workflow of the Outsourcing Services Negotiation System, starting from user registration and login, followed by service request submission, negotiation, submission review, invoice handling, and finally the logout process. Each diagram reflects the collaborative interactions between clients, administrators, and operations managers, supported by systematic validation and feedback mechanisms. Through this modeling, the system's dynamic behavior is clearly mapped out, providing a solid foundation for both implementation and future improvements.

c. Sequence diagram

To describe how components of the system interact dynamically over time, a sequence diagram is presented to complement the previously discussed activity diagrams. This diagram illustrates the order of messages exchanged between actors (Client, Admin, and Operations Manager) and system modules (such as AuthService, RequestHandler, NegotiationModule, etc.) during the execution of key business processes, including login, service request submission, negotiation, invoice handling, and logout.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

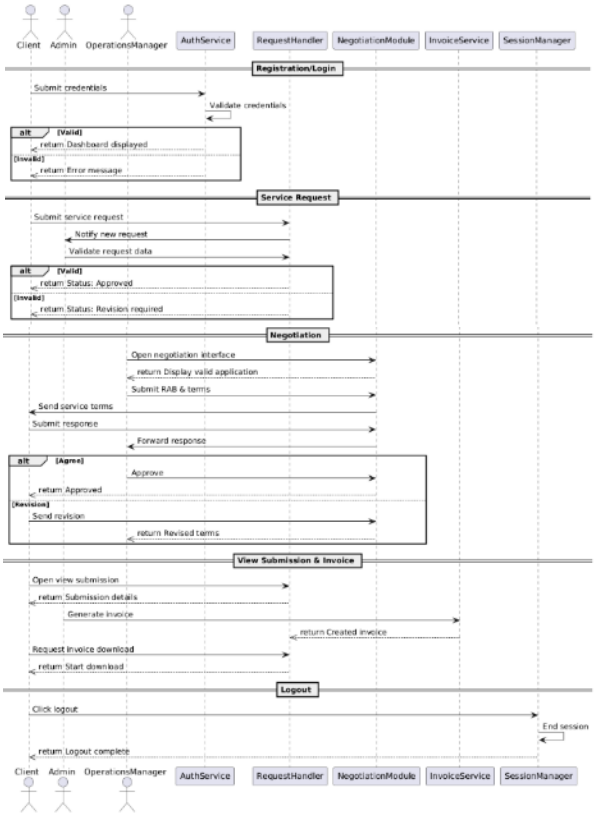

Figure 8. Sequence diagram

This sequence diagram clarifies the interaction flow and role distribution among various actors and services within the system architecture. By modeling time-based message exchanges, it helps ensure that the system design aligns with the intended business logic and supports a clear implementation roadmap for developers.

d. Class diagram

The following class diagram is used to illustrate the static structure of the Outsourcing Services Negotiation system, including class relationships and the primary responsibilities of each entity. The diagram includes eight main classes, such as User, AuthService, SessionManager, RequestHandler, NegotiationModule, and InvoiceService. The User class serves as a central entity representing different user roles client, admin, and operations manager distinguished through the role attribute. Relationships among classes are annotated with multiplicity to show the cardinality of interactions.

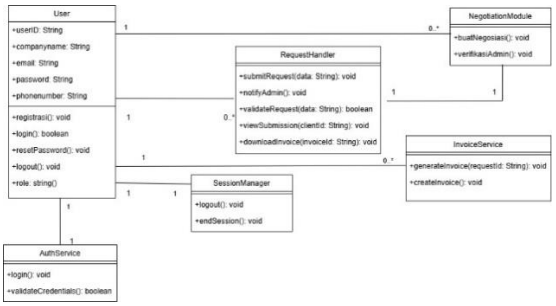

Figure 9. Class diagram

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

This class diagram facilitates the understanding of the systems structure from an object-oriented programming perspective. With clearly defined responsibilities and multiplicity relationships between classes, it provides a solid foundation for implementing modular and structured system logic. The model also supports future development through a flexible role based access control approach.

e. Mockup design

In the design of the Outsourcing Services Negotiation system, the user interface (UI) plays a vital role in supporting effective and intuitive user interaction. To visualize and validate the system functionality, mockups were developed to reflect the intended user experience. This article presents three primary screens: the login page, dashboard, and negotiation form.

First, the Login mockup illustrates the entry point into the system. It includes fields for username/email and password. The layout is intentionally simple and responsive to accommodate all user roles (client, admin, and operations manager).

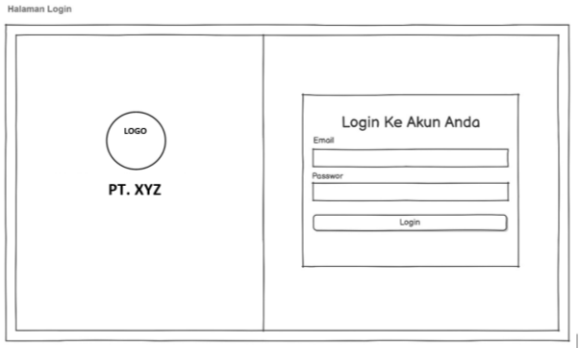

Figure 10. Login form design

Second, the Dashboard interface provides a role based navigation experience. The design supports direct access to essential system functions with minimal visual clutter.

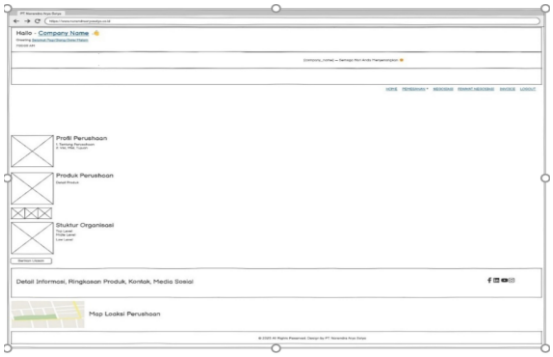

Figure 11. Dashboard design

Third, the Negotiation Form mockup facilitates interaction between the client and the operations manager. This interface allows users to review service details, submit revisions, or approve proposed terms. Fields for inputting service scope, duration, and cost are presented clearly to support transparent and efficient decision-making.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

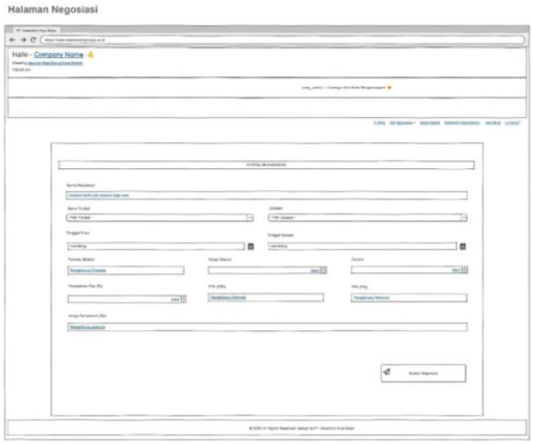

Figure 12. Negotiate design

These three views were selected for presentation because they represent the core system workflow while demonstrating the application of user-centered design principles.

3. Implementating the System
The following is an implementation of the main interface of the system along with a brief description of its functionality:

a. Login page
The login page is used by registered users and admins to access the system by entering their email address and password. This login system has been equipped with a standardized validation and security system.

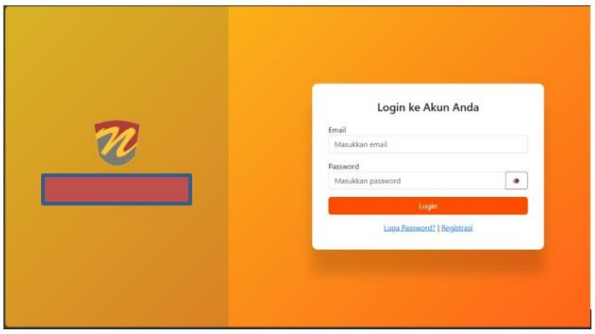

Figure 13. Login page

b. Negotiate page
The Negotiation page is used to propose a price that can be approved or rejected, for the service, the duration of the contract, and other specific needs before the service order is executed.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

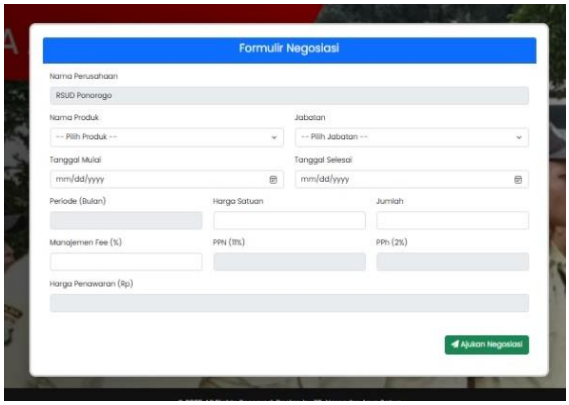

Figure 14. Negotiate page

4. Testing Stage
To ensure that all system functionalities operated in accordance with user requirements, the system underwent Black Box Testing. This method evaluates the output of each function based on given inputs, without inspecting the internal code structure. Testing was conducted on key modules including login, service request, negotiation, invoice generation, and logout. In addition to qualitative verification, quantitative performance metrics were recorded to strengthen the evaluation. These included execution time for each function and error rates during repeated trials. Testing was performed three times for each module under varied but valid input conditions.

Table 4. Black box testing

Test Case	Input	Expected Output	Actual Output	Execution Time (seconds)	Error Rate (%)
Login	Valid username/ password	Access granted	Access granted	1.2	0
Service Request	Valid service data	Service request stored	Service request stored	3.5	0
Negotiation	Price and duration proposal	Negotiation terms updated	Negotiation terms updated	5.0	0
Invoice Generation	Approved negotiation details	Invoice generated	Invoice generated	2.8	0
Logout	Logout request	Session ended	Session ended	1.0	0

Based on the testing results conducted using the Black Box Testing method, all core features of the system have demonstrated behavior consistent with the expected specifications. Each module, from user authentication to invoice download, functioned correctly and no anomalies were found under different input scenarios. This indicates that the system successfully meets the functional requirements and is suitable for deployment in an operational environment. The successful test outcomes also reflect that the system development process was carried out systematically and adhered to sound software engineering principles.

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

5 CONCLUSION

Manual negotiation process as outlined in the research objectives and problem statement in the introduction. The system was designed to transform the negotiation process between clients, administrators, and operations managers into a systematic, interactive, and traceable digital workflow. The adoption of XP allowed for a flexible, collaborative, and iterative development process, enabling rapid adaptation to evolving user needs.

Through an object-oriented programming approach, the system achieves a modular structure and effective role-based access management. The resulting prototype integrates a user-friendly login page, dashboard, and negotiation form, all designed to enhance usability. Black Box Testing demonstrated that all system functions operated according to specifications with zero anomalies, confirming that the system is ready for operational deployment. This outcome aligns with the study's objective to improve

Both efficiency and transparency in outsourcing service negotiations. Despite these achievements, the research has limitations. The evaluation was conducted within a controlled testing environment, without performance benchmarking against comparable systems or stress-testing under high-load conditions. Furthermore, user feedback was limited to internal stakeholders, which may restrict generalizability.

For future work, it is recommended to expand the evaluation scope to include larger-scale operational environments and diverse user groups, integrate quantitative performance benchmarks, and explore comparative studies with alternative Agile methodologies. System enhancements could include API integration for digital signatures, automated notifications via email or WhatsApp, and advanced security measures such as end-to-end encryption and detailed audit trails. These developments would not only strengthen the system's practical value but also contribute to academic knowledge on the application of Agile methods in digital negotiation systems.

REFERENCES

Fatoni, F., & Irawan, D. (2019). Implementasi Metode Extreme Programming dalam Pengembangan Sistem Informasi Izin Produk Makanan. Jurnal Sisfokom (Sistem Informasi Dan Komputer), 8(2), 159–164. https://doi.org/10.32736/sisfokom.v8i2.679

Hou, C., Lu, M., Deng, T., & Shen, Z. J. M. (2021). Coordinating project outsourcing through bilateral contract negotiations. Manufacturing & Service Operations Management, 23(6), 1466–1483. https://doi.org/10.1287/msom.2020.0911

Huang, H., Hu, M., Kauffman, R. J., & Li, T. (2021). The power of renegotiation and monitoring in software outsourcing: Substitutes or complements? Information Systems Research, 32(4), 1085–1107. https://doi.org/10.1287/isre.2021.1026

Lamada, M., Bakry, A., Ifani, A. Z., & Khaerunnisa, K. (2023). Development of Web-Based Project Tender Documents Application Using Extreme Programming Methods. Elinvo (Electronics, Informatics, and Vocational Education), 7(2), 101–111. https://doi.org/10.21831/elinvo.v7i2.49863

Putri, S. J., Galih, D., Putri, P., Hayuhardhika, W., & Putra, N. (2024). Analisis Komparasi pada Teknik Black Box Testing (Studi Kasus: Website Lars). Journal of Internet and Software Engineering, 5(1), 23-28. https://doi.org/10.22146/jise.v5i1.9446

Setyawati, E. (2021). Web-Based Management Information System for Services Development: A Literature Review. International Journal of Current Science Research and Review, 04(03), 175-185. https://doi.org/10.47191/ijcsrr/V4-i3-05

Shrivastava, A., Jaggi, I., Katoch, N., & Bhardwaj, A. (2021). A systematic review on extreme programming. Journal of Physics: Conference Series, 1969(1), 012046. https://doi.org/10.1088/1742-6596/1969/1/012046

e-ISSN: 2655-867X
p-ISSN: 2655-8661
Volume 7 No. 2 August 2025

Siska Narulita, Ahmad Nugroho, & M. Zakki Abdillah. (2024). Diagram Unified Modelling Language (UML) untuk Perancangan Sistem Informasi Manajemen Penelitian dan Pengabdian Masyarakat (SIMLITABMAS). Bridge: Jurnal Publikasi Sistem Informasi Dan Telekomunikasi, 2(3), 244–256. https://doi.org/10.62951/bridge.v2i3.174

Sitio, S. L. M., Tanu, D. Y., Solihin, S., Saifudin, A., & Desyani, T. (2023). Pengujian Blackbox pada Website Open Jurnal Universitas Pamulang Menggunakan Teknik Cause-Effect Relationship Testing. Jurnal Informatika Universitas Pamulang, 8(1), 102–106. https://doi.org/10.32493/informatika.v8i1.26885

Verhoef, P. C., Broekhuizen, T., Bart, Y., Bhattacharya, A., Qi Dong, J., Fabian, N., & Haenlein, M. (2021). Digital transformation: A multidisciplinary reflection and research agenda. Journal of Business Research, 122, 889–901. https://doi.org/10.1016/j.jbusres.2019.09.022

Widhiastuti, S., Permata, R., & Hendrastuty, N. (2023). Rancang bangun sistem informasi kepegawaian berbasis website dengan menggunakan metode Extreme Programming pada kantor kelurahan X. Jurnal Teknologi dan Sistem Informasi, 10(10), 291-301.

Z. A. W. Sugandi, Y. A. Nugraha, S. N. Anam, and I. Darmayanti, 'Implementasi Konsep Pemrograman Berorientasi Objek Dalam Aplikasi Pembukuan Keuangan Penjual Jus Buah Menggunakan Bahasa Pemrograman Java', Jurnal Ilmiah IT CIDA, vol. 8, no. 1, p. 1, 2022, doi: 10.55635/jic.v8i1.154.